Purdue University
Purdue e-Pubs

Department of Computer Science Technical Reports
Department of Computer Science

1975

A Study of Program Locality and Lifetime Functions

Peter J. Denning

Kevin C. Kahn

Report Number:
75-148

A STUDY OF PROGRAM LOCALITY AND LIFETIME FUNCTIONS

Peter J. Denning
Kevin C. Kahn

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

CSD TR 148

revised

September 1975

Work reported herein supported in part by NSF Grant GJ-41289

A STUDY OF PROGRAM LOCALITY AND LIFETIME FUNCTIONS[1]

Peter J. Denning[2]
Kevin C. Kahn[2]

Abstract: A program model can be regarded as decomposible into two main parts. The macromodel captures the phase-transition behavior by specifying locality sets and their associated reference intervals (phases). The micromodel captures the reference patterns within phases. A semi-Markov model can be used at the macrolevel, while one of the simple early models (such as the random-reference or LRU stack) can be used at the microlevel. This paper shows that, even in simplest form, this type of model is capable of reproducing known properties of empirical lifetime functions. A micromodel alone, without a macromodel, is incapable of doing so.

[1]Work reported herein was supported in part by NSF Grant G5-41289.

[2]Authors' address: Department of Computer Sciences, Purdue University, West Lafayette, IN 47907.

Technical Report TR-148

Memorandum No. PD75:7

1. INTRODUCTION

The stochastic nature of computer system performance variables derives from randomness in the way programs drive the operating system. Therefore, an understanding of program behavior is the basis for an understanding of system behavior. An important manifestation of program behavior is the lifetime function [BeK69]. It gives the mean processor execution interval between page faults as a function of the mean memory space allocated to a program. This function can be used in a queueing network to obtain estimates of mean throughput and response time, of the computer system modelled by the network, for various values of the degree of multiprogramming. Such estimates can be quite good; see [Bra74, Cou75, Den75, Mun75].

If one is interested only in estimating the means of performance variables, then empirical lifetime functions (or empirically verified approximations to them) are sufficient to parameterize a queueing network. If, however, one is interested in more general questions -- such as the effect on response time when the instantaneous demand for main memory exceeds the supply -- descriptions of program behavior more detailed than lifetime functions are required.

There is wide agreement that, to be able to match observed behavior, program models should embody what is known as "locality". This behavior is better described as phase-transition behavior [Den72, Fer74, MaB75, SpD72], or as "regime" behavior [FGS75]. The idea is that a program's execution is regarded as a sequence of "phases", each of which is a sequence of references over an associated "locality set" (of pages). Experiments based on sampling a reference string and noting the pages referenced in each sample interval have amassed considerable indirect evidence of this behavior (e.g., [Bry75, HaG71, Rod71]). The most striking direct evidence has been given recently by Madison and Batson [MaB75], who define a phase as a maximal interval in which LRU stack distance does not exceed i (for given i) and every one of the i top stack objects is referenced at least once. The definition and associated experiments show that phases (and associated locality sets) can be nested within larger phases (and associated locality sets) for several levels. The "outermost" level tends to be characterized by long phases with transitions between nearly disjoint locality sets. (Interestingly, the outermost phase tended not to be the program's entire execution interval.) The inner levels have shorter phases and overlapping sets. The innermost level of interest depends on the system: phases whose lifetimes are short compared to the paging time are of no interest. Thus the practical question is not whether phase transition behavior exists, but how to model it.

A few experiments to test very simple program models have been reported in the literature, the most successful of which depend on the "LRU stack model" (see [AKS73, SpD72, ShT72, CoD73]). None of these models contains an explicit concept of phase-transition behavior. All have been found unsuitable as descriptions of long term program behavior. These simple models are incapable of explaining all the observed properties of lifetime funtions. For example, Spirn [Spi73] and Arvind et al. [AKS73] reported that, for working set parameter T, the LRU stack model would fit the fault rate at T with average relative error 30% and more. The LRU model predicts that the LRU fault rate will be better than WS at almost all memory allocations [Spi73], contradicting empirical observations that WS may be significantly better over significant ranges of memory allocation [Bar73, Bar75, PrF73].

The experiments reported herein are a next step in the evolution of models for intrinsic program mechanisms. They are controlled experiments designed to test the hypothesis that a very simple model based on variable localities and phase transitions is sufficient to explain at least the gross behavior of programs. Our experiments confirm this hypothesis. The model to be described is capable of generating reference strings whose lifetime functions exhibit the properties observed in empirical lifetime functions. The following pages describe the lifetime properties observed from past experiments, the model we used, the experimental results, and the limitations of the model.

2. LIFETIME FUNCTIONS

2.1. Definitions

For a given policy of memory management and space constraint x, the <u>lifetime function</u> L(x) specifies the mean number of references in a given program between requests to secondary memory devices -- in our context, L(x) is the mean virtual time between page faults [BeK69, CFL72, Den74, Gel73, Sal74]. (The expiration of a lifetime interval corresponds to a transition from the processor to the paging device.) If the memory policy is a <u>fixed-space policy</u>, the space constraint x means that the program's resident set (set of pages in main memory) contains exactly x pages; for such a policy x is a directly controllable parameter. If the memory policy is a <u>variable-space policy</u>, the space constraint x means that the program's resident set contains an average of x pages, the average being measured over virtual time; for such a policy there is usually some parameter whose setting can be used to cause x to vary over some suitable range.

To state this more precisely, let r(k) denote the number of pages in the resident set of a given program just after the kth reference, where k=1,2,...,K and K is the reference string length. For a fixed space policy, r(k) = x for all k. For a variable space policy,

$$x = \frac{1}{K}\sum_{k=1}^{K} r(k) \ . \qquad (1)$$

In our experiments, we measured the page fault rate function f(x) and set $L(x) = 1/f(x)$. (This formula is exact if a page fault is assumed to occur at time K.)

As a representative fixed space policy, we chose the least recently used (LRU) procedure. As a representative variable space policy, we chose the moving window working set (WS) procedure. These were chosen not only because they are typical, but because their fault-rate functions can be measured efficiently [DeG75].

2.2. Properties

Experiments or analyses reported in the literature reveal a number of distinguishing properties of lifetime functions. To be consistent with our knowledge of programs, a model of program behavior must at the least be able to reproduce them.

Figure 1 shows a typical lifetime function, L(x). Since x = 0 would imply a fault on every reference, L(0) = 1. Lifetime functions ordinarily have a convex region for small x and a concave region for large x. The point x_1 is the point of maximum slope -- i.e., an inflection point -- separating the two regions. The point x_2 is the "knee" of the curve, defined as the tangency point of a ray eminating from L(0)=1; its significance will be discussed shortly. Belady [BeK69] reported that the convex region can be approximated by a function cx^k where typically $1.5<k<2.5$; these results were verified by Spirn [Spi73]. (Actually $1+cx^k$ would yield a slightly better approximation.) Larger values of k are usually considered as indicating greater degrees of cyclic reference patterns within program phases. Summarizing:

> Property 1: A lifetime function typically has the convex/concave shape. The convex region is approximated by cx^k for some (c,k), and larger k-values will tend to be associated with more cyclic programs.

Studies by Bard [Bar73, Bar75] and by Prieve and Fabry [Pri73, PrF73] have shown the existence of significant regions in which the WS lifetime exceeds the LRU. Indirect evidence of the same type was found by Chu and Opderbeck [ChO72] in the observation that WS space-time was significantly less than LRU space-time over the range of parameter choices of interest. When the distribution of locality size is bimodal, several crossover points may be observed (see below). Figure 2 shows a typical case, where x_0 is the first crossover point. Summarizing:

> Property 2: For a given reference string, the WS lifetime function will tend to exceed that of LRU, often significantly, for wide ranges of memory allocations.

Given that one accepts the existence of phase-transition behavior in programs, one can regard memory management policies as <u>locality</u>

<u>estimation procedures</u> seeking to detect locality sets associated with phases whose length is comparable to or larger than the page fault service time. Each assigns the resident set at a given time to be a container of the locality set associated with the subject program's current phase. An <u>ideal estimator</u> has three properties: a) the resident set is always a subset of the current locality set, b) at a transition the resident set contains only the pages in common to the old and new locality sets, and c) page faults occur only for first references to <u>entering pages</u>, i.e., those pages of the new locality set not also in the old.* A phase is at "nesting level" k if it is contained within k-1 other phases. Take k=1 as the outermost or uppermost level [MaB75]. Define these quantities:

- H_k, mean phase holding time,
- M_k, mean number of pages entering locality at a transition,
- R_k, mean overlap, i.e., number of pages remaining in locality at a transition,
- m_k, mean locality set size, and
- u_k, mean resident set size of ideal estimator.

Our definitions imply that

(2) $$u_k \leq m_k = R_k + M_k.$$

The lifetime function of the ideal estimator will satisfy

(3) $$L(u_k) = H_k/M_k.$$

(A proof is given in Appendix A.) The lifetime function of the ideal estimator would be expected to flatten out for $x>u_1$: once the outermost locality sets are kept resident, increases in space beyond u_1 are not expected to produce as much improvement in lifetime as increases in space approaching u_1. In other words, we expect the knee x_2 of the lifetime curve (Figure 1) of the ideal estimator to occur approximately at u_1.

The working set with window of size T is not an ideal estimator because it retains resident old locality pages for up to T references after a transition. As long as T is short enough so that the window rarely contains more than one transition, but long enough to observe all locality pages contained in a phase, the working set will produce the same paging rate as the ideal estimator. (See also [Den75].) Therefore, corresponding to each space u_k of the ideal estimator is a larger space w_k of the working set at which $L(w_k)=H_k/M_k$, where the excess w_k-u_k represents the overestimate at transitions. In particular, the knee of the working set lifetime will occur at lifetime of approximately H_1/M_1. From now on, we shall drop the subscript 1 and assume that H and M refer to the outermost level of phases. The foregoing argument is summarized:

<u>Property 9</u>: At the knee x_2 of the WS lifetime curve, $L(x_2)$ is approximately H/M, where H is the mean holding time of

---

*If each locality set page is referenced at least once in a window T contained anywhere in a phase, the optimal policy VMIN will behave as an ideal estimator [PrF75, Den75].

outermost phases and M is the mean number of pages entering locality at transitions.

For the same reasons, a similar property will hold for the LRU lifetime.

Using an assumption that changes in locality set size follow a Gaussian process with mean m and standard deviation σ, Coffman and Ryan [CoR72] studied the relative efficiency of fixed and variable space policies. They showed that variable space policies are always better than fixed, but the differences may be slight if the fixed resident set is at least $m+2\sigma$. Translated into out context, these results have the following interpretation. The standard deviation of the number of pages entering at a transition is σ. The variable space policy of Coffman and Ryan is an ideal estimator of outermost phases; it has, therefore, mean resident set m and lifetime of H/M. The (imperfect) estimator in a fixed space policy (e.g., LRU) requires space x_2 to achieve the same lifetime. For comparable performances, therefore, we expect $x_2=m+k\sigma$ for some small constant $k\geq 1$. Summarizing:

<u>Property 4</u>: When the distribution of locality size is Gaussian, the knee x_2 of a fixed space policy lifetime curve should differ from the mean locality size m by an amount proportional to σ, the standard deviation of the number of pages entering the locality set at a transition between outermost phases.

3. THE MODEL AND THE EXPERIMENTS

Our objective was limited: modeling phase-transition behavior at the outermost level [MaB75] using as few parameters as possible. To quantify phase-transition behavior, four factors must be specified:

1. The distribution of holding time in each locality set -- i.e., the durations of the phases;
2. The process by which the program chooses new locality sets at phase transitions;
3. The extent to which locality sets of adjacent phases overlap; and
4. The process by which the program generates references from within a given locality set.

Factors 1-3 pertain to the <u>macromodel</u> (i.e., to the phase-transition behavior) while factor 4 pertains to the <u>micromodel</u> (i.e., to the reference pattern within each phase). This model is more complex than previous models (such as the LRU stack or independent reference models [CoD73, ShT72, SpD72, Spi73]) which are in our context merely possible micromodels.

A semi-Markov chain can be used to describe the macromodel over a given collection of locality sets $S_1,\ldots,S_n$. When the chain is in state i, the program is assumed to be in a phase referencing S_i. A holding time distribution $h_i(t)$ describes the probability that a phase over S_i lasts for t references; $\bar{h}_i$ is its mean. A transition matrix $[q_{ij}]$ gives the probabilities that S_j is the next locality set after S_i. Let $\{Q_i\}$ denote the equilibrium distribution of the matrix $[q_{ij}]$ and define

$$p_i = \frac{Q_i\bar{h}_i}{P}, \quad P = \sum_{i=1}^{n} Q_i\bar{h}_i . \tag{4}$$

Since p_i denotes the fraction of time that S_i is the locality set, we shall refer to $\{p_i\}$ as the <u>observed locality distribution</u>. If l_i is the size of S_i, then the mean and variance of this distribution are respectively,

$$m = \sum_{i=1}^{n} p_i l_i, \quad \sigma^2 = \sum_{i=1}^{n} p_i l_i^2 - m^2 . \tag{5}$$

As has been stated, the limited objective of our experiments was testing the ability of simple phase-transition models to reproduce known properties of program behavior. For this reason we chose to specify the macromodel with as few parameters as possible:

1. The holding time distribution was state independent, i.e., $h_i(t) = h(t)$ for all i, with mean $\bar{h}$.
2. The type of the observed locality distribution, together with its mean m and standard deviation σ, were given. This was used to derive a distribution $\{p_i\}$ over locality sizes $\{l_i\}$, which in turn was used to determine the locality sets $\{S_i\}$.
3. At a phase transition, S_i is entered with probability p_i. In other words, $q_{ij}=p_j$ for all i, and in particular $Q_j=p_j$.

These choices require only 2n+1 parameters $(\bar{h},p_1,\ldots,p_n,S_1,\ldots,S_n)$ rather than at least $2n+n^2$ for the full semi-Markov chain. This approach has the additional advantage of allowing us to investigate the effect of the observed locality distribution on the results. It has the limitation that the probability of entering a locality set of size j depends on j alone; the effect of this limitation will be discussed later.

Because this formulation permits an unobservable model transition from S_i to S_i, the observed holding time in S_i is a geometric sum of model holding times; its mean is $\bar{h}/(1-p_i)$, which is larger than $\bar{h}$. The mean holding time over all phases will be observed as

$$H = \bar{h} \sum_{i=1}^{n} \frac{p_i}{1-p_i} . \tag{6}$$

This value of H, rather than the smaller value $\bar{h}$, must be used in evaluating the results with respect to Property 3 of lifetime functions.

Table I summarizes our choices for the model parameters. The common holding time distribution was exponential. A few preliminary experiments showed that other choices of this distribution with the

Factor	Choices			
1. Holding time distribution	Exponential, mean $\bar{h}$=250			
2. Locality size distribution				
a. Type	Uniform	Gamma	Normal	Bimodal
b. Mean m	30	30	30	see
c. Standard deviation σ	5,10	5,10	5,10	Table II
3. Transition matrix $[q_{ij}]$	from locality distribution(see text)			
4. Mean overlap R	none (R=0)			
5. Micromodel	cyclic, sawtooth, random			
6. Memory policy	LRU, WS			

Table I: Choices of factors.

			Modes					
			Normal $N_1(v)$			Normal $N_2(v)$		
Number	m	σ	w_1	m_1	σ_1	w_2	m_2	σ_2
1	30	5.7	.50	25	3.0	.50	35	3.0
2	30	10.4	.50	20	3.0	.50	40	3.0
3	30	10.1	.33	16	2.0	.67	37	2.0
4	30	7.5	.33	20	2.5	.67	35	2.5
5	30	10.0	.60	22	2.1	.40	42	2.1

$$\text{Bimodal}(v) = w_1 N_1(v) + w_2 N_2(v)$$

w_1, w_2, are mode weights

Table II: Bimodal distributions.

same mean produced no significant effect on the results. The mean of the distribution was chosen as $\bar{h}=250$; for the locality size distributions used, this produced H values ranging from 270 to 300. Values of H observed in practice are likely to be an order of magnitude larger than this [Gra75, HaG71, Rod71]. An experiment using a more realistic value verified that the only observable effect of changing $\bar{h}$ is a rescaling of lifetime on the vertical axis. Since none of the conclusions would be affected by a change in time scale, we chose the smaller (and less expensive) value of $\bar{h}$.

For the locality size distribution, we chose discrete approximations to uniform, normal, gamma, and "bimodal"; these choices represent the range of symmetric and skewed types observed in practice [Bry75, Rod71]. The range of locality sizes covered by each distribution was partitioned into n intervals, for n ranging from 10 to 14 depending on the complexity of the distribution. We chose p_i to be the occupancy probability of the ith interval and l_i to be its midpoint. The locality set S_i is a set of l_i distinct page names. To simplify comparisons among the results, we chose mean m=30 pages for all distributions. This choice was large enough to permit studying a range of coefficients of variation (ratio σ/m). Using different means while holding other factors fixed would do little more than rescale L(x) along the horizontal axis.

Table II summarizes the bimodal distributions. Reflecting observations [Bry75, GhK73, Rod71], each is approximated as the superposition of two normal distributions.* They range from symmetric (nos. 1 and 2), to high-skewed (nos. 3 and 4) and low-skewed (no. 5). Their means and standard deviations were computed according to equations (5) and are shown in the left columns of the table.

To approximate transitions among nearly disjoint locality sets in the outermost phases, we made the simple assumption of mutually disjoint locality sets. In other words, the mean overlap (R) was zero and the mean number of entering pages (M) was the mean locality size (m). The principal effect of increasing the mean overlap (R) while holding all other factors fixed would be a vertical expansion of the lifetime function (e.g., since the point x_2 does not depend on R, the knee would vary vertically as $L(x_2)=H/(m-R)$); this effect would not offset comparisons among lifetime functions. We confirmed this reasoning with a few experiments.

For micromodels, we chose cyclic, sawtooth, and random reference patterns. Each locality set was stored as a list, e.g.,

---

*A result by Denning and Schwartz [DeS72, CoD73] shows that asymptotic uncorrelation of references will produce normally distributed working set size. That bimodal distributions are observed shows that this property does not always hold.

$$S_i = S_i[0],\ldots,S_i[l_i-1].$$

An index pointer j was used to select the next reference: as long as S_i is the current locality set, $0 \le j < l_i$. The cyclic micromodel simply chooses the next value of j as $(j+1) \bmod l_i$. This model corresponds to reference patterns for which LRU will generate one page fault per reference whenever the memory space x is less than l_i, i.e., a worst case for LRU. The sawtooth micromodel sweeps the pointer j up and down the list, generating sequences of j-values like

$$0,1,\ldots,l_i-1,l_i-1,\ldots,1,0.$$

This model corresponds to patterns for which LRU will be optimal or nearly so [DeG75]. The random micromodel uses a uniform random number generator to choose the next value of j. It is intended as a simple representation of a stochastic reference string.

In our initial runs we did not try an LRU stack micromodel because it would have required additional parameters (the stack distance frequencies). As the experiments progressed, we observed that changes in the micromodel affected primarily the convex region of the lifetime function, these effects being small for the WS lifetime. In other words, the significant effects arise from the macromodel: a more complex micromodel would not have altered our conclusions. For this reason no experiments with LRU stack micromodels were conducted.

Table I specifies 11 choices of the distribution and 3 micromodels, or a total of 33 program models. For each of these program models, as well as for some additional ones needed to confirm intuitions and conclusions, we generated one reference string encompassing K=50000 references (about 200 phase transitions). Within each experiment, the procedure was simply to repeat the following until K references were generated: choose a locality set S_i with probability p_i and holding time t according to h(t); then generate t references from S_i using the micromodel. As each reference was generated, LRU stack distance and interreference interval counts were updated; at the conclusion of each run, this data was used to construct the lifetime curves for LRU and WS, using well known methods [CoD73, DeG75].

4. RESULTS

Our discussion of results is divided into two parts: verifying first that the model is consistent with the known Properties 1-4 of lifetime functions, then identifying patterns of model behavior which may turn out to be characteristic of programs. Figures 3-7 are representative of a much larger set of plots obtained for each experiment.

4.1. Consistency

Consistent with Property 1, all lifetime functions exhibited the convex/concave shape. However, the LRU lifetime function for the bimodal distributions tended to have two inflection points for $x<x_2$, which were correlated with the positions of the modes. In trying to fit cx^k to the convex regions we found that k approximately 2 gave best fits for the random micromodel, which compares favorably with values observed in practice [BeK69]. In contrast, k=3 or larger was required for the cyclic or sawtooth micromodels. This is consistent with the empirical observation that real programs' referencing behaviors tend to be randomized, and not so simply predictable.

Consistent with Property 2, the WS lifetime function was higher than the LRU over a significant range (Figure 3 is typical). Except in the cyclic micromodel (for which LRU is poor), we observed that the first crossover point x_0 was always at least m. At larger values of σ we observed also that $x_0<x_2$(LRU), while at the smaller values of σ we observed that $x_0 \approx x_2$(LRU). This is consistent with the empirical observation that WS has little or no advantage over LRU except where the coefficient of variation σ/m is large [CoR72, Pri73, PrF73].

Consistent with Property 3, all curves displayed a knee at approximately the point $L(x_2)=H/m$; since H ranged from 270 to 300, the lifetime values ranged from 9 to 10 at the knees.

Consistent with Property 4, the knee of the LRU lifetime curve satisfied the relation $x_2-m=k\sigma$ for $1<k<1.5$. In all cases $(x_2-m)/1.25$ was a good estimate of σ. Additional experiments with $\sigma=2.5$ verified this conclusion. The quality of this approximation deteriorated for the bimodal distributions.

4.2. Other Conclusions

A number of other patterns were exhibited by these experiments. Some are consistent with expectations about program behavior. Others are new, and will require further study to determine if in fact they are properties of programs.

Pattern 1: An $x_1=m$ Property. In every experiment we observed the striking property that WS lifetime curve had inflection point $x_1=m$, to within the precision of the experiments (Figure 4 is typical). This property was also observed for LRU with two notable exceptions: 1) It did not hold for the cyclic micromodel. 2) For the bimodal distributions there tended to be two inflection points, corresponding to but smaller than the modes of the distribution; the change in slope at the first point was larger in direct proportion to the weight of the smaller mode (w_1 in Table II). We could find no simple hypothesis to explain the $x_1=m$ property -- we offer it as a prediction of the model subject to future investigation.

Pattern 2: Independence of WS Lifetime from Higher Moments of the Locality Distribution. Aside from dependence on the mean already mentioned, the WS lifetime curves showed no significant dependence on either the standard deviation (σ) or the form of the locality size distribution. (Figure 5 is typical.)

Pattern 3: Dependence of LRU Lifetime on Higher Moments of the Locality Distribution. The LRU lifetime curve exhibited strong dependence on the locality size distribution. (Figure 5 is typical.) As noted earlier, the knee x_2 is approximately $m+1.25\sigma$; this approximation deteriorated for the bimodal distributions. For the bimodal distributions, lifetimes in the concave region grew larger in correlation with increasing weight of the smaller mode; Figure 6 shows that many tended to exhibit a second crossover with WS lifetime curve.

Pattern 4: Dependence on the Micromodel. The knees $L(x_2)$ of all lifetime curves tended to be H/m independent of the micromodel. The shape of the WS lifetime is (often much) less sensitive to the micromodel than is the LRU (Figure 7 is typical). However, if the WS lifetime curve points $(x,L(x))$ are labeled with the window values $(T(x))$ that produced them, a dependence is observed:

(7) WS: $T(x)(\text{cyclic}) < T(x)(\text{sawtooth}) < T(x)(\text{random})$;

a factor of 2 between the extremes of (7) was typical. The reason is that a larger window is required to achieve a working set of given size x in a randomized reference pattern. We observed also that the WS difference x_2-x_1 (which is approximately x_2-m) increases with randomness in the micromodel:

(8) WS: $x_2(\text{cyclic}) < x_2(\text{sawtooth}) < x_2(\text{random})$.

The reason is that x_2-m measures the WS overestimate at phase transitions which, by virtue of inequalities (7), increases with randomness. In contrast, LRU lifetime curves exhibit strong dependencies on micromodel. As expected, LRU was worst on the cyclic micromodel (Figure 6). The x_2 inequalities for LRU are the reverse of those above (8) of WS; the reason is that, for $x>m$, the probability of stack distance not exceeding x is largest for the random micromodel and smallest for the cyclic.

5. LIMITATIONS

The generality of our conclusions must be evaluated in light of the limiting effects of four properties of the model and experiments. The first limitation, inherent in the model, is the state independent holding time distribution ($h(t)=h_i(t)$ for all states i). The primary effect is to make the fraction of time locality set S_i is current the same as the equilibrium distribution of the transition matrix. In preliminary experiments on real reference strings, Graham [Gra75] has found that, with a state independent holding distribution, a semi-Markov model of empirical working set size accurately reproduces the observed WS lifetime. He observes empirically that a small fraction of the working set sizes account for a high fraction of the equilibrium occupancy probability; among these dominant sizes the differences among holding time distributions are not major. Thus, the use of a single holding time distribution in a model does not appear to be a serious limitation.

A second limitation, also inherent in the model, is the use of a single locality probability n-vector $[p_i]$ rather than a full nxn transition matrix $[q_{ij}]$. We have in effect replaced the matrix by its equilibrium distribution. As noted, this endows the model with the property that the probability of entering state i is independent of past or future states. Since the convex region of the lifetime curve is dominated by the micromodel, and the effect of the macromodel becomes progressively more noticeable starting from the inflection point (x_1), this limitation of the macromodel would be significant only for space constraints well into the concave region of the lifetime curve.

A third limitation, of the experiment but not the model, is our use of disjoint locality sets in adjacent phases -- i.e., the mean number R of pages remaining across a transition is zero. Our choice of R=0 was mostly a matter of convenience inspired by our interest in outermost phases. It is easy to construct an instance of the model in which R>0. The only difficulty is that there seems to be no simple way of estimating R from an empirical lifetime function. (Recall that we can estimate m, and H if R is known.)

A fourth limitation, again of the experiment but not the model, is our omission of the LRU stack model as a micromodel. This was again a matter of convenience to keep the number of parameters small -- including an LRU stack distance distribution over k pages would require k additional parameters. Our experience with micromodels suggests that an LRU stack model would not affect the shape of the convex region very much; however preliminary results by Graham [Gra75] show that this micromodel will cause WS lifetime triplets (x,L(x),T(x)) -- in which T(x) is the WS window required to achieve a working set size of x -- to correspond closely with those of an empirical WS curve. In other words, the additional complexity of the LRU stack micromodel would be warranted only if we required the model triplets to correspond closely with the empirical ones.

Prior work has shown the LRU stack model with independent stack distances to be the best of a class of simple models, none of which is based on phase-transition behavior. Phase-transition models, even with micromodels simpler than the LRU stack model, appear better as reproducers of known lifetime properties. As with the other micromodels, the LRU stack model must be subjected to a phase-transition superstructure to be capable of accurately reproducing properties of empirical lifetime functions.

6. CONCLUSIONS

The experiments corroborate the following as a general description of the WS lifetime function. The properties of the convex region will generally be dominated by the micromodel because the WS window contains a transition only a small portion of the time. (In particular, approximations cx^k for k approximately 2 and $x \le x_1 = m$ will obtain for micromodels exhibiting randomized, rather than highly deterministic, patterns.) The properties of the concave region of a WS lifetime function will generally be dominated by the macromodel because the WS window contains at least one phase transition most of the time. For the region $x_1 \le x \le x_2$, the WS window contains no more than one transition with any significant probability, and so the lifetime curve behavior in this region depends primarily on the properties of single transitions (to wit, the curve passes smoothly from $L(x_1) \simeq cm^k$ to $L(x_2) \simeq H/M$). For $x > x_2$, the WS window contains multiple phase transitions with increasingly significant probability, so that the remainder of the concave region depends significantly on the details of the transitions among sequences of locality sets. The macromodel limitations can thus affect the WS lifetime significantly only for $x > x_2$.

Relating the properties of LRU lifetime functions to macro and micromodels is not as straightforward: they depend on complex interactions among all the factors.

Parameterizing an instance of the model from empirical LRU and WS lifetime curves is not difficult: 1) The mean locality size is taken as $m = x_1$. 2) The standard deviation of locality size is estimated as $\sigma = (x_2 - m)/1.25$, where x_2 is the knee of the LRU lifetime. 3) Assuming that adjacent localities tend to be disjoint (as in outermost phases), the WS value $mL(x_2)$ is an estimate of mean holding time H. (In general, H is estimated from $(m-R)L(x_2)$, where R is the mean overlap across a transition; in our context, no method of estimating R is known.) Since the limitations of the model are most pronounced in the region $x > x_2$, it is likely that an instance of the model so parameterized would agree well with observations for the range $x \le x_2$. A more complex macromodel --e.g., one with full transition matrix -- would be required if the agreement in the concave region were poor.

The general conclusion is that a simple phase-transition model is capable of reproducing known properties of program lifetime functions over at least the range of memory constraints up through the knee of the lifetime curve. Without some form of phase-transition behavior the model would be incapable of reproducing known properties. If n locality sets are used in the model, only 2n+1 parameters are required. A model of more parameters may be required for better reproduction of lifetime properties for large memory constraints [Gra75].

ACKNOWLEDGEMENT

We are grateful especially to Alan Batson for much assistance in interpreting these experiments and understanding their limitations; to G. Scott Graham for information on the progress of his own experiments with empirical models; and to those unnamed souls, the referees, for posing so many hard questions.

APPENDIX A

For an ideal estimator at some level with space constraint u, paging arises precisely from phase transitions. Consider a sequence of N phases of holding times $\{h_j\}$. Let m_j be the number of pages in the jth locality but not in the j-1st; note that $F = m_1+\ldots+m_n$ is the total number of page faults. Let l_{jk} be the length of the kth lifetime interval in phase j $(k=1,\ldots,m_j)$; note that h_j is the sum of l_{jk} over $k=1,\ldots,m_j$. Applying the definitions,

$$H = \frac{1}{N}\sum_{j=1}^{N} h_j = \frac{L}{N} = \frac{F}{N}\frac{L}{F} .$$

By assumption, L(u) is L/F, where L is the total of all lifetimes. Observing that M=F/N is the mean number of pages entering the resident set at each transition, we obtain L(u)=H/M.

REFERENCES

[AKS73] Arvind, R. Y. Kain, and E. Sadeh. "On Reference String Generation Processes." Proc. 4th ACM Symp. on Operating Systems Principles (October 1973), 80-87.

[Bar73] Bard, Y. "Characterization of Program Paging in a Time Sharing Environment." IBM J. of R. & D. 17, 5 (Sept. 1973), 387-393.

[Bar75] Bard, Y. "Application of the Page Survival Index (PSI) to Virtual Memory System Performance." IBM J. of R. & D. 19, 3 (May 1975), 212-220.

[BeK69] Belady, L. A., and C. J. Kuehner. "Dynamic Space Sharing in Computer Systems." Comm. ACM 12, 5 (May 1969), 282-288.

[Bra74] Brandwajn, A. "A Model of a Time Sharing Virtual Memory System Solved Using Equivalence and Decomposition Methods." Acta Informatica 4 (1971), 11-47.

[Bry75] Bryant, P. "Predicting Working Set Sizes." IBM J. of R. & D. 19, 3 (May 1975), 221-229.

[CFL72] Chamberlin, D. D., S. H. Fuller, and L. Y. Lin. "A Page Allocation Strategy for Multiprogramming Systems with Virtual Memory." Proc. 4th ACM Symposium on Operating Systems Principles (October 1973), 66-72.

[ChO72] Chu, W. W. and H. Opderbeck. "The Page Fault Frequency Replacement Algorithm." Proc. AFIPS Conf. 40 (1972 FJCC), 597-609.

[CoD73] Coffman, E. G. Jr. and P. J. Denning. Operating System Theory, Prentice-Hall (1973).

[CoR72] Coffman, E. G. Jr. and T. A. Ryan Jr. "A Study of Storage Partitioning Using a Mathematical Model of Locality." Comm. ACM 15, 3 (March 1972), 185-190.

[Cou75] Courtois, P. J. "Decomposability, Instabilities, and Saturation in Multiprogramming Systems." Comm. ACM 18, 7 (July 1975), 371-376.

[Den72] Denning, P. J. "On Modeling the Behavior of Programs." Proc. AFIPS Conf. 40 (1972 SJCC), 937-944.

[Den74] Denning, P. J. "Comments on a Linear Paging Model." Proc. ACM SIGMETRICS Symposium (October 1974), 34-48.

[Den75] Denning, P. J. "The Computation and Use of Optimal Paging Curves." Technical report TR-154, Computer Sciences Department, Purdue University (June 1975).

[DeG75] Denning P. J. and G. S. Graham. "Multiprogrammed Memory Management." Proc. IEEE: Special Issue on Interactive Computer Systems, (June 1975), 924-939.

[DeS72] Denning, P. J. and S. C. Schwartz. "Properties of the Working Set Model." Comm. ACM 15, 3 (March 1972), 191-198.

[Fer74] Ferrari, D. "Improving Locality by Critical Working Sets." Comm. ACM 17, 11 (November 1974), 614-620.

[FGS75] Freiberger, W. F., U. Grenander, and P. D. Sampson. "Patterns in Program References." IBM J. of R. & D. 19, 3 (May 1975), 230-243.

[Gra75] Graham, G. S. "Program Behaviour and Memory Management." Ph.D. Dissertation, Computer Sciences Department, Purdue University, in progress.

[Gel73] Gelenbe, E. "The Distribution of a Program in Primary and Fast Buffer Storage." Comm. ACM 16, 7 (July 1973), 431-434.

[GhK73] Ghanem, M. Z. and H. Kobayashi. "A Parametric Representation of Program Behavior in a Virtual Memory." IBM T. J. Watson Research Center Report (Sept. 1973).

[HaG71] Hatfield, D. J. and J. Gerald. "Program Restructuring for Virtual Memory." IBM Sys. J. 10, 3 (1971), 168-192.

[MaB75] Madison, W. and A. Batson. "Characteristics of Program Localities." Proc. 5th ACM Symp. on Operating Systems Principles (November 1975), to appear.

[Mun75] Muntz, R. R.. "Analytic Modeling of Interactive Systems." Proc. IEEE: Special Issue on Interactive Computer Systems (June 1975), 946-953.

[Pri73] Prieve, B. G. "Page Partition Replacement Algorithm." Ph.D. Thesis, Dept. Elec. Engrg. and Computer Science, University of California, Berkeley (December 1973).

[PrF73] Prieve, B. G. and R. S. Fabry. "Evaluation of a Page Partition Replacement Algorithm." Technical Report, Bell Laboratories, Naperville, Illinois (October 1973).

[PrF75] Prieve, B. G., and R. S. Fabry. "VMIN -- an Optimal Variable Space Replacement Policy." Proc. 5th ACM Symposium on Operating Systems Principles (November 1975), to appear.

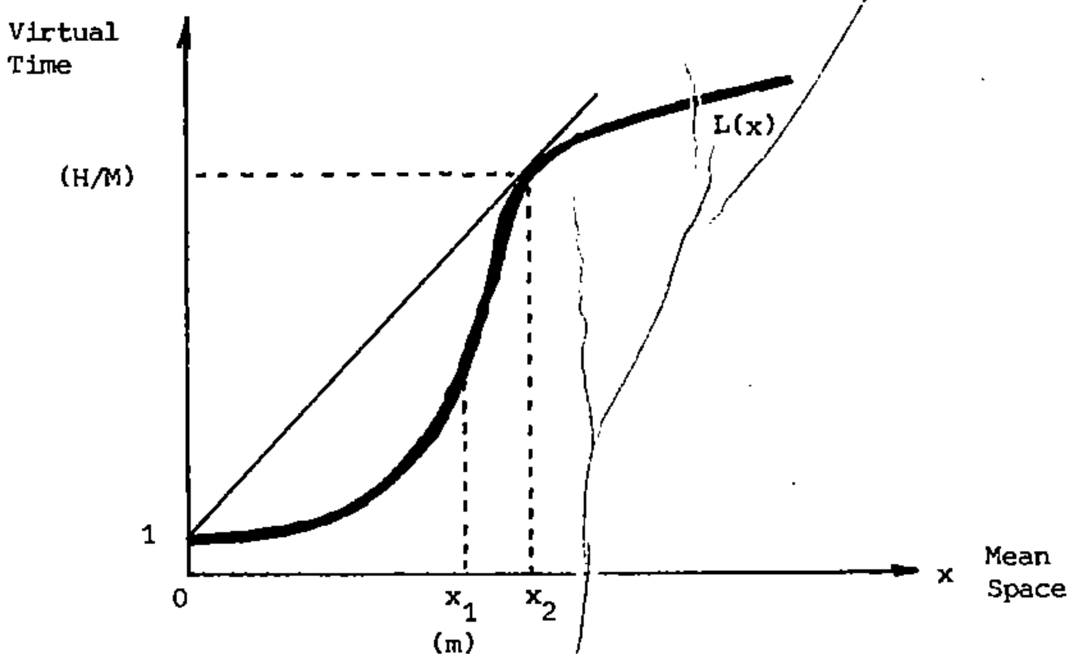
[Rod71] Rodriguez-Rosell, J. "Experimental Data on How Program Behavior Affects the Choice of Scheduler Parameters." Proc. 3rd ACM Symposium on Operating System Principles (October 1971), 156-163.

[Sal74] Saltzer, J. H. "A Simple Linear Model of Demand Paging Performance." Comm. ACM 17, 4 (April 1974), 181-185.

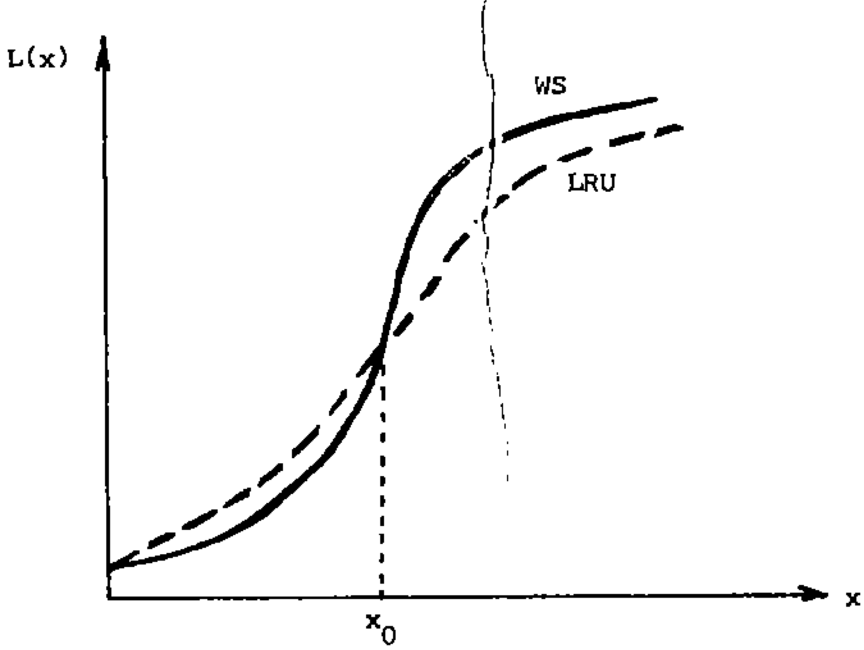
[SpD72] Spirn, J. R. and P. J. Denning. "Experiments in Program Locality." AFIPS Conf. Proc. 41 (1972 FJCC), 611-621.

[Spi73] Spirn, J. R. "Program Locality and Dynamic Memory Management." Ph.D. Thesis, Dept. Elec. Engrg., Princeton Univ. (March 1973).

[ShT72] Shedler, G. S. and C. Tung. "Locality in Page Reference Strings." SIAM J. Computing 1, 3 (September 1972), 218-241.



<u>Figure 1</u>. A lifetime curve.



<u>Figure 2</u>. Comparison of lifetime curves.

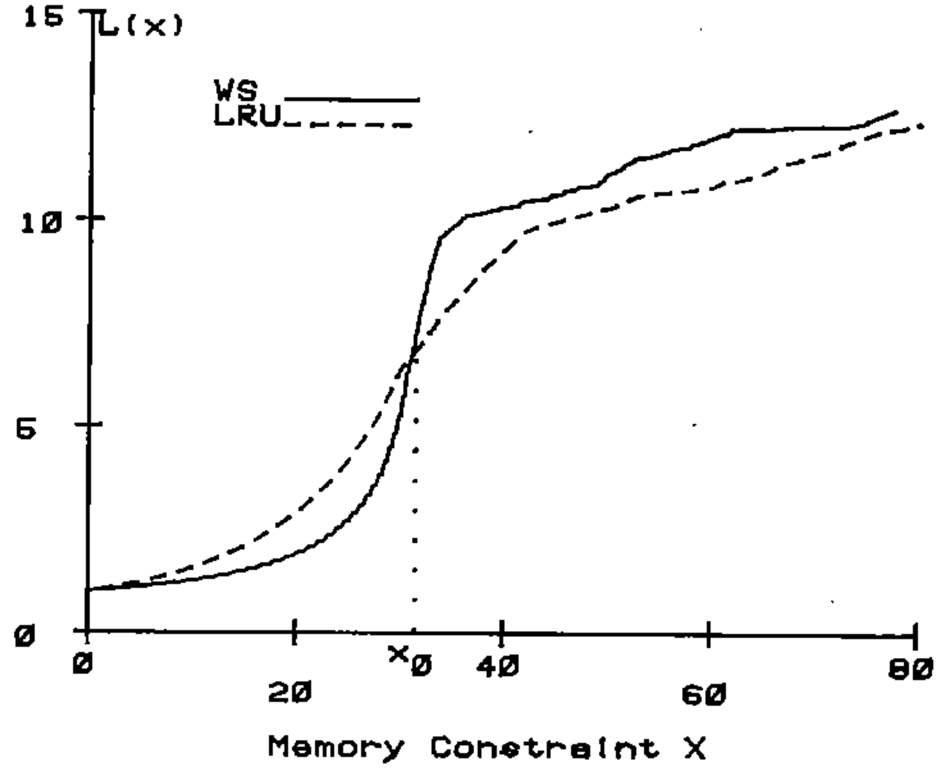


Figure 3. Normal dist. - sawtooth micromodel - std. dev.=10.

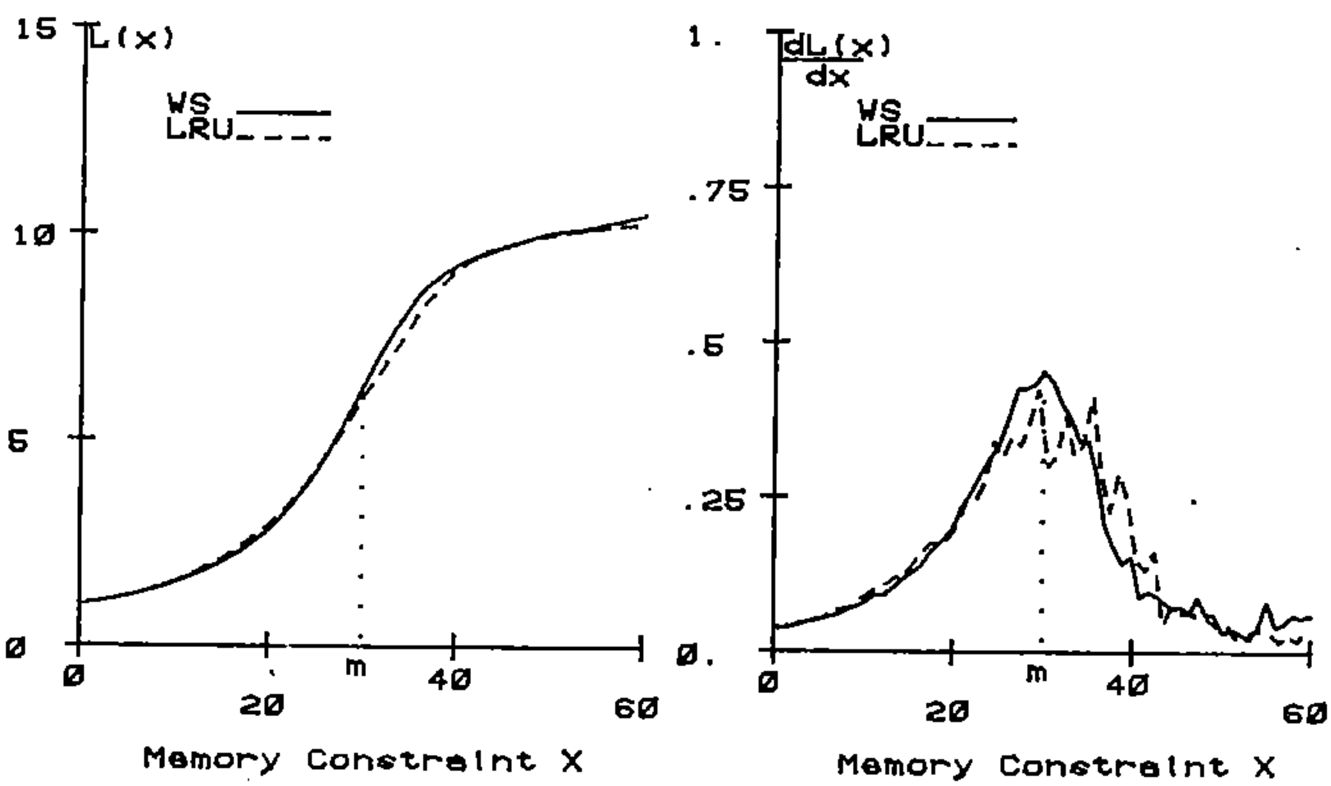


Figure 4. Gamma dist. - random micromodel - std. dev.=10.

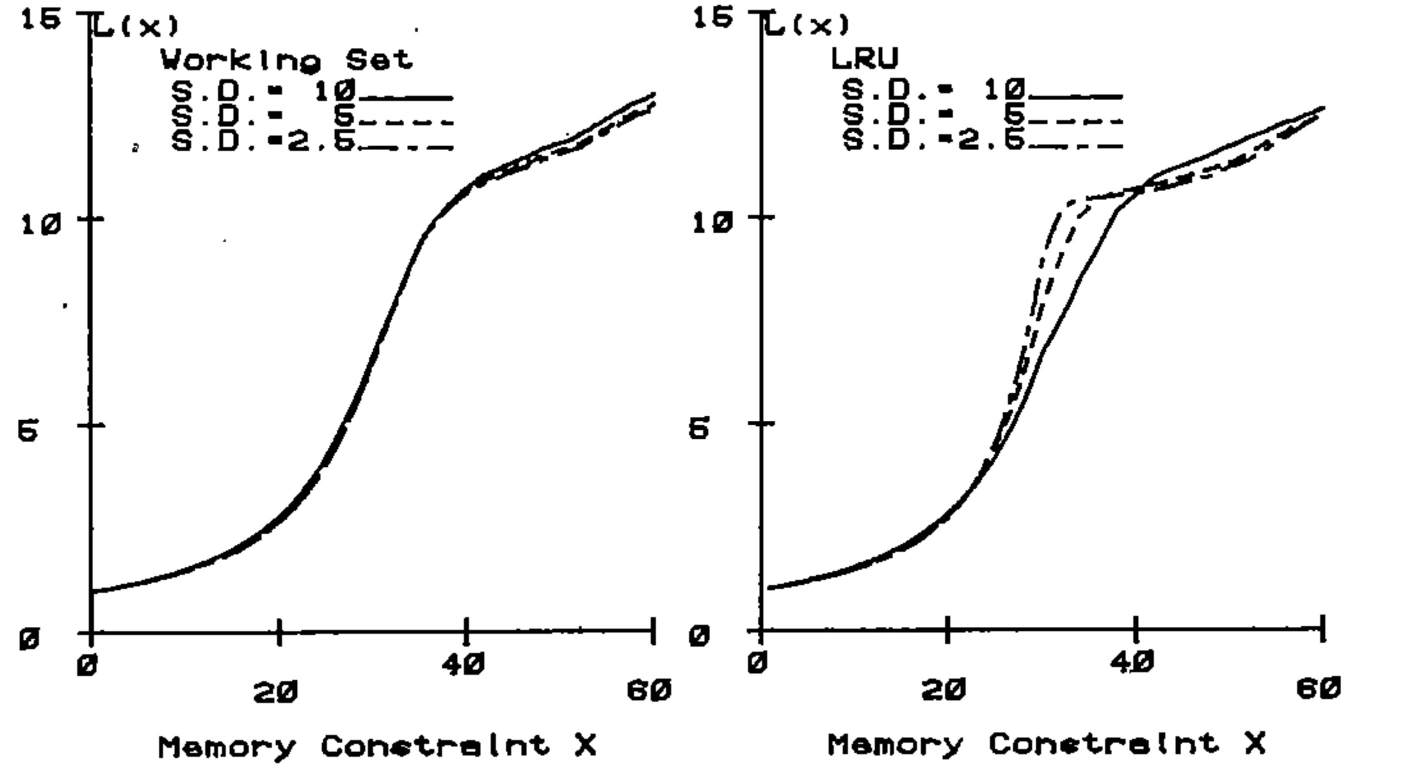


Figure 5. Effect of variance (Normal dist. - random micro.).

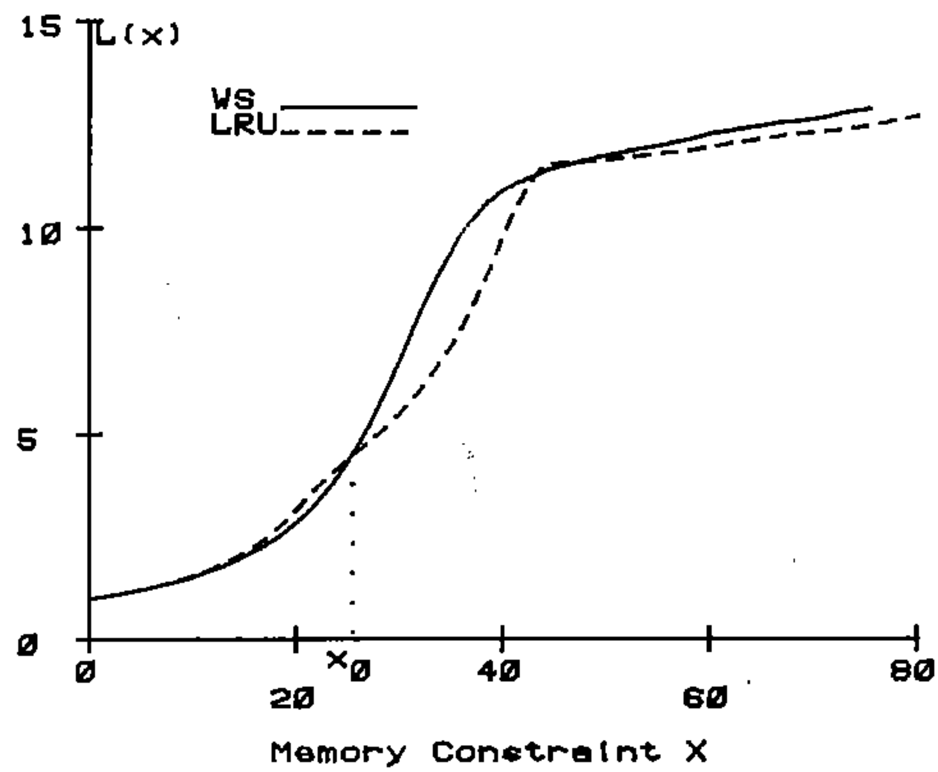


Figure 6. Bimodal 5 - random micromodel - std. dev.=10.

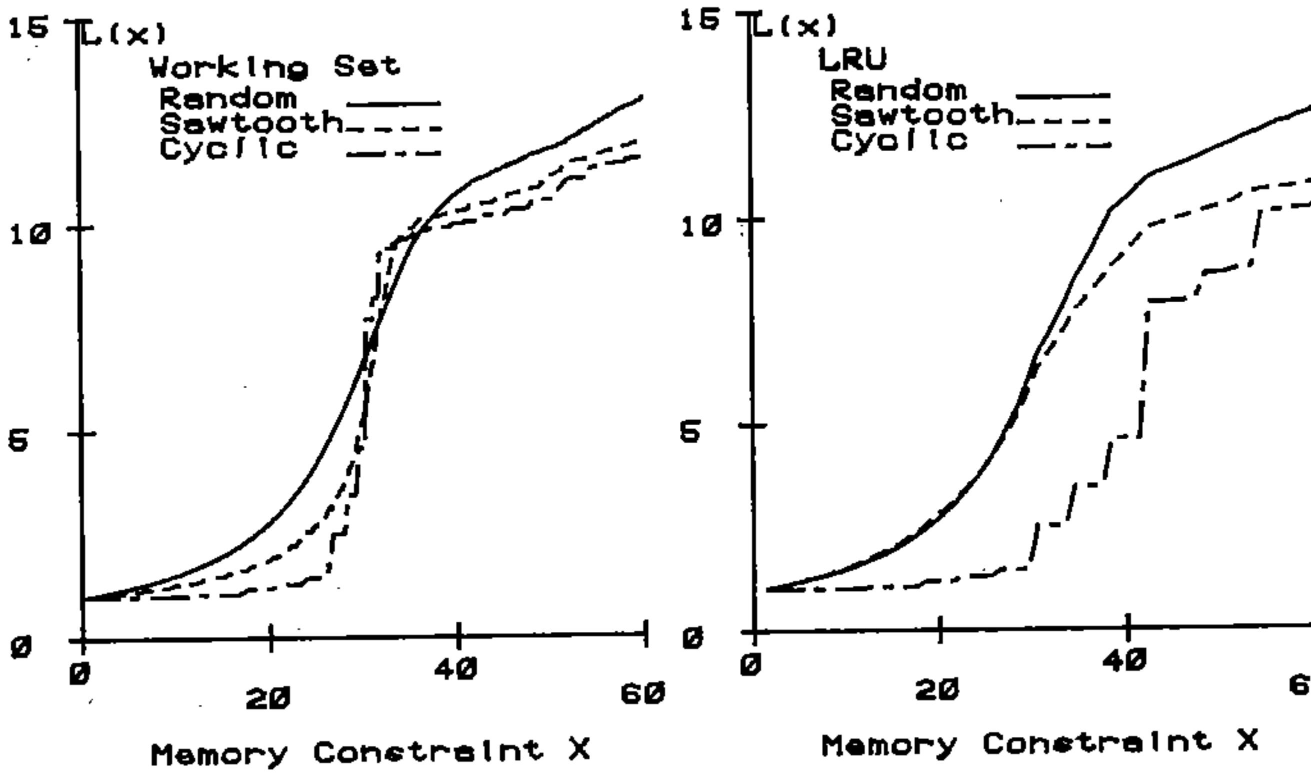


Figure 7. Comparison of micromodels (Normal dist. - dev.=10).